Article │ Received 8 January 2025; Accepted 10 March 2025; Published 14 March 2025
https://doi.org/10.55092/pcs20250008

Applications of case-based methods in teaching formal language and automata theory

Wei Wang, Yangxuan Cheng*, Guanglu Zhou, Xin Sun and Ping Yu

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

* Correspondence author; E-mail: yangxuancheng@stu.hit.edu.cn.

Abstract: Formal language and automata theory is an essential core course for outstanding computer workers. This course, with the characteristics of abstraction and formalization, has rigorous theoretical proofs, stronger constructiveness and the fundamental models' establishment and their properties. To help students acquire a better understanding of the theoretical knowledge, enhance their problem-solving abilities and improve their application skills, this paper, based on a case study on fall detection and alarm systems for elderly people, introduces finite-state automata abstract definitions, thereby stimulating students' learning enthusiasm.

Keywords: case-based methods; formal language and automata theory; the finite-state machines; the elderly fall detection and alarm systems

1. Introduction

Computer workers possess four fundamental abilities: computational thinking skills, algorithm design and analysis skills, programming and implementation skills, and abilities of cognition, analysis, design and application in terms of computer hardware and software systems. Computational thinking skills comprise abstract thinking abilities and logical thinking abilities; formal language and automata theory can cultivate such skills through a gradual teaching process in the course delivery.

In order to support the teaching of formal language and automata theory courses, Rodger *et al.* from Duke University in the United States developed the interactive graphic teaching software JFLAP over a period of 15 years [1]. The feature of this software is that it can visually display the connotations of many basic concepts and theorems in the course in the form of charts [2]. In order to change the teacher centered and exam-oriented teaching mode and reflect student-centered self-directed learning. Huai Libo and others adopt a discussion based teaching method, paying more attention to the personalized needs of students in terms of discussion content and form, and integrating personalized educational ideas [3]. In order to enhance students' self-learning ability and promote their independent thinking, Pei Zhishu and others proposed a heuristic teaching reform plan, encouraging students to actively participate in the learning process, improving their self-learning ability and creativity, enhancing their problem-solving ability and learning motivation [4]. However, the course content itself is relatively abstract, as is seen when explaining finite-state machine concepts-how to understand what constitutes an 'automaton' poses a significant challenge. To facilitate students' comprehension, this paper employs a case study on fall detection and alarm systems for elderly people to introduce abstract definitions of finite-state machines.

In home environments, robots can track elderly individuals' movements, implement proactive tracking and possess fall detection capabilities as well. The robot continuously monitors video screens from cameras to detect falls in real-time; upon detecting a fall, it immediately notifies children or hospitals through mobile apps [5–7]. This system primarily consists of the robot and smartphone. The robot is an intelligent small car equipped with laser radar and depth cameras, enabling SLAM mapping indoors and human pose estimation for tracking elderly individuals and detecting falls. If a fall occurs, the alarm function is triggered. On the smartphone, two mobile apps were developed: one for elders (allowing them to summon the robot) and another for children (enabling reception of emergency alert messages, such as fall notifications with accompanying photos), facilitating timely assistance.

The introduction of engineering cases and the infiltration of engineering experience enable students to have a deeper understanding and mastery of theoretical knowledge, transform textbook knowledge into practical skills, cultivate students' engineering project development concepts, accumulate engineering practical experience, and also provide talent reserves for enterprises. Rich engineering practical experience can provide students with clearer career choices and development opportunities.

2. Finite state automata design for elderly fall detection and alarm system

The robot and smartphone operate independently yet interconnectedly when providing services, with mobile apps designed for elders and children relying on information provided by the robot. Therefore, we will design finite-state machines [8–10] separately for each client-side platform running on both the robot and smartphone.

The finite-state machines for the robot, elderly-focused APP, and offspring-focused APP are denoted as $M_1 = (Q_1, \Sigma_1, \delta_1, q_0, F)$, $M_2 = (Q_2, \Sigma_2, \delta_2, q_0, F)$, and $M_3 = (Q_3, \Sigma_3, \delta_3, q_0, F)$, respectively. All three finite-state machines belong to the NFA, $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a non-empty set of finite states; Σ is an alphabet set; $\delta(\delta\colon Q \times \sum \rightarrow 2^Q)$ is a transition function set such that for any $(q, a) \in Q \times \Sigma$, $\delta(q, a) = \{p1, p2, \ldots, pm\}$ indicates that M can choose to move from state q to one of the states p1, p2, …, or pm upon reading character a and moving the read head one step to the right along the input string. The initial state is denoted as q_0; F represents the set of terminal states.

2.1. Finite automaton for robot

The robot's possible states during its operational process are represented as the state set Q_1 of M_1, which consists of ten distinct states: {'Start Robot', 'Connect Network', 'APP Pairing', 'Image Capture', 'Track User', 'Pose Estimation', 'Fall Confirmation', 'Notify Child', 'Notify Hospital', 'Close Robot'}.

The processing functions executed by the robot during its operational process are represented as the input alphabet Σ_1 of M_1. Σ_1 = {'Transition from initial state to network selection state, complete network connection (input a1)', 'Complete pairing with mobile APP and robot (input b1)', 'Start robot's installed depth camera and deploy lightweight model and detection algorithm (input c1)', 'Initiate user tracking process by capturing depth data through camera, maintaining safe distance between robot and the elderly, and continuous tracking (input d1)', 'Continuously capture images to obtain consecutive frames (input e1)', 'Use lightweight model for skeletonization of human body (input f1)', 'Maintain persistence of user's skeletal modeling by continuously capturing images (input g1)', 'Estimate pose using posture estimation algorithm to judge potential fall situation, and grab current image. To avoid false alarms, confirmation can be sent to the user, waiting for response (input h1)', 'If no one responds or the user answers "need help", generates assistance information and notifies children (input i1)', 'If no one responds or the user answers "need help", generates assistance information and notifies hospitals (input j1)', 'Notify child complete/Complete the procedure of notifying children; if the robot's running low on power, close robot (input k1)', 'Notify hospital complete/Complete the procedure of notifying hospitals; if the robot's running low on power, close robot (input l1)', 'The elder does not need robotic assistance, re-enter image capture state (input m1)', 'The user does not need robotic tracking, command the robot to leave or the robot's power is insufficient, and it ends service (input n1)'}.

The robot's finite state machine starts in the initial state q_0 = 'Start Robot'; The terminal state set F is defined as {'Close Robot'}, which represents the final states of the robot's finite state machine when it has completed its service provision due to user command or insufficient power. The transition function δ_1: $Q_1 \times \Sigma_1 \rightarrow Q_1$ maps the current state q $\in Q_1$ to a new state q' $\in Q_1$ based on the input symbol $\sigma \in \Sigma_1$. The mapping relationship is shown below and is depicted in Figure 1:

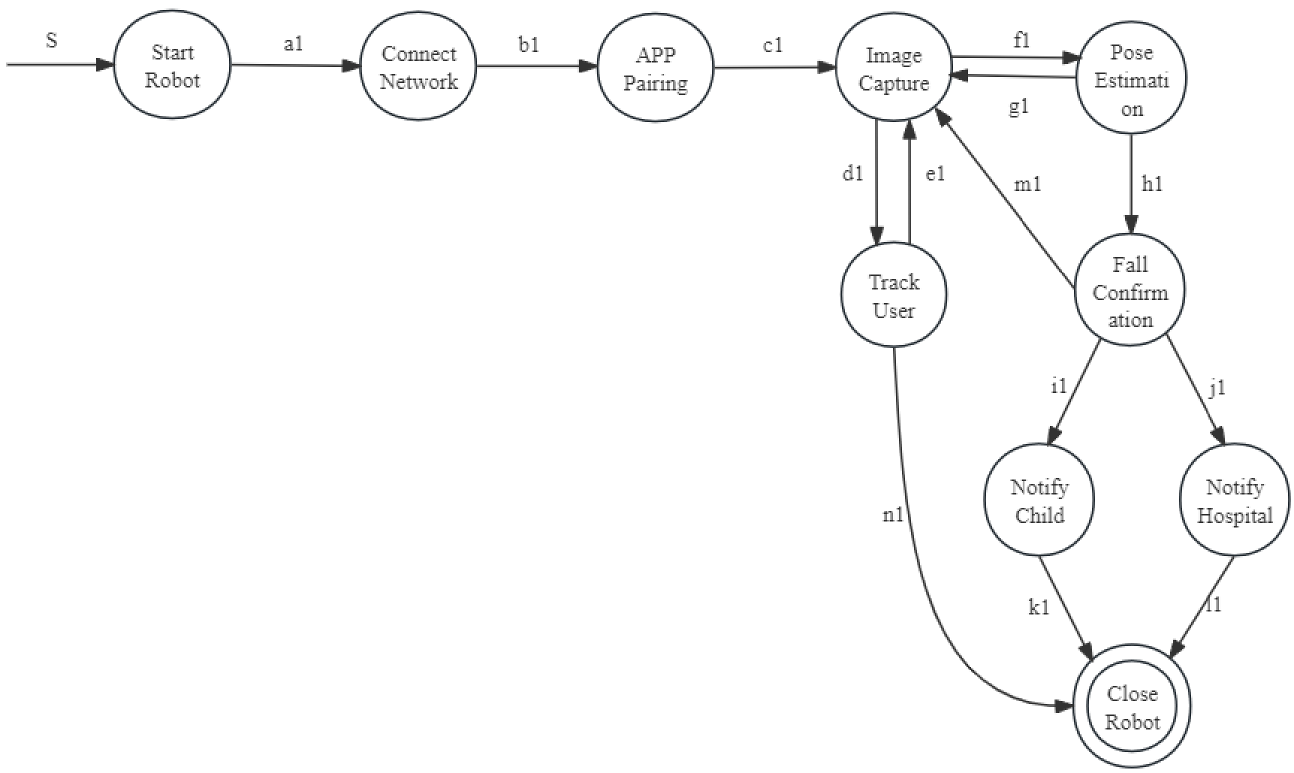

Figure 1. Finite state machine (FSM) diagram of robot M_1's state transition.

δ_1("Start Robot", a1) = "Connect Network";
δ_1("Connect Network", b1) = "APP Pairing";
δ_1("APP Pairing", c1) = "Image Capture";
δ_1("Image Capture", d1) = "Track User";
δ_1("Track User", e1) = "Image Capture";

δ_1(“Image Capture”, f1) = “Pose Estimation”;
δ_1(“Pose Estimation”, g1) = “Image Capture”;
δ_1(“Pose Estimation”, h1) = “Fall Confirmation”;
δ_1(“Fall Confirmation”, i1) = “Notify Child”;
δ_1(“Fall Confirmation”, j1) = “Notify Hospital”;
δ_1(“Notify Child”, k1) = “Close Robot”;
δ_1(“Notify Hospital”, l1) = “Close Robot”;
δ_1(“Fall Confirmation”, m1) = “Image Capture”;
δ_1(“Track User”, n1) = “Close Robot”.

2.2. Finite automaton for senior-focused mobile app

The mobile elderly version of the fall detection and alarm system’s Android application (hereinafter referred to as M_2) may exhibit the following states during its operation, which are represented by the finite state machine M_2: Q_2 = {“Start APP”, “Connect Network”, “Voice Service”, “Connect Robot”, “Robot Summoned”, “Robot Departure”, “Close APP”} with a total of 7 states.

The input alphabet Σ_2 for M_2 represents the processing functions executed by the elderly version during its operation, which are as follows: Σ_2 = {“Start the APP and enter the connection network state (input a2)”, “Following the user’s manual activation of voice services, the elderly version’s speech recognition function is activated (input b2)”}, {“Following connection to the network, the elderly version can manually locate robot information through Bonjour service and connect the robot (input c2)”}, {“After connecting to the network, the elderly version can use its speech recognition interface to attempt to connect the robot (input d2)”}, {“Upon successful pairing with the robot, the elderly version’s voice services are activated, summoning the robot to the user’s location (input e2)”}, {“Upon successful pairing with the robot, the elderly version’s voice services are activated, commanding the robot to depart from the user’s location (input f2)”}, {“After the robot departs, the user can terminate the process or exit the program to end use of the elderly version APP (input g2)”}, {“After the robot is summoned, the user can terminate the process or exit the program of the elderly version APP (input h2)”}.

The elderly-version APP has a finite state machine with an initial state q_0 = “Start APP”. The termination states are defined as F = {“Close APP”}, which represents the service closed or exit the program. The transition function $\delta_2 : Q_2 \times \Sigma_2 \rightarrow Q_2$ for the elderly version APP is mapped as follows and is depicted in Figure 2:

δ_2(“Start APP”, a2) = “Connect Network”;
δ_2(“Connect Network”, b2) = “Voice Service”;
δ_2(“Connect Network”, c2) = “Connect Robot”;
δ_2(“Voice Service”, d2) = “Connect Robot”;
δ_2(“Connect Robot”, e2) = “Robot Summoned”;
δ_2(“Connect Robot”, f2) = “Robot Departure”;
δ_2(“Robot Departure”, g2) = “Close APP”;
δ_2(“Robot Summoned”, h2) = “Close APP”.

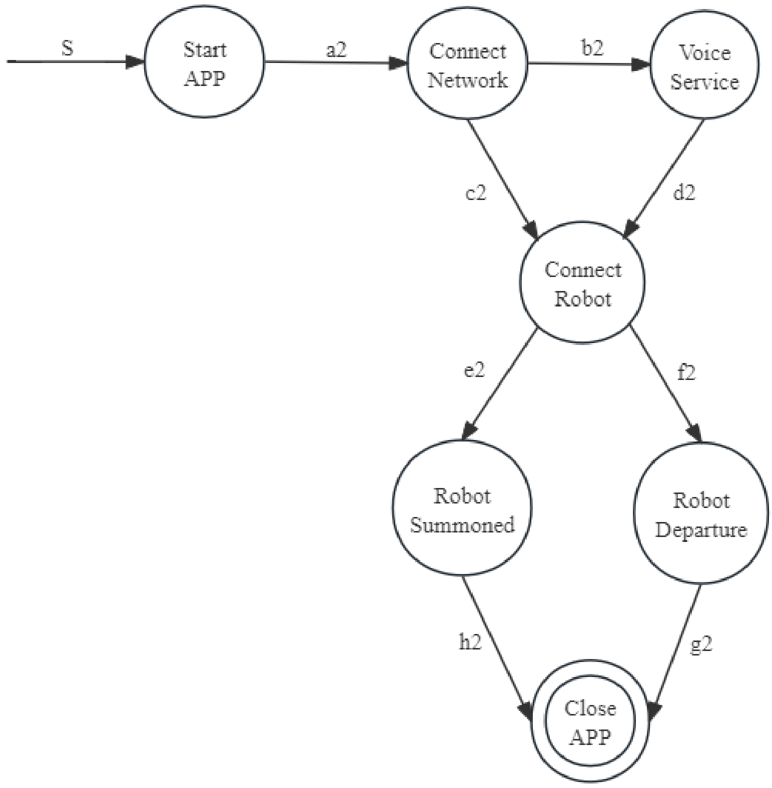

Figure 2. Finite state machine (FSM) diagram of senior-focused mobile app M_2's state transition.

2.3. Finite automaton for offspring-focused app

The offspring-focused mobile APP of the fall detection and alarm system's Android application (hereinafter referred to as M_3) may exhibit the following states during its operation, which are represented by the finite state machine M_3: Q_3 = {"Start APP", "Bind Family Member", "Family Home Page", "Fall Image", "Alarm SMS", "Self-Management"} with a total of 6 states.

The offspring-focused APP executes processing functions during its operation, which are represented by the input alphabet Σ_3 for M_3. Σ_3 = {"After starting the child version APP, if no family member has been bound previously, it directly enters the binding interface from the initial state (input a3)"}, {"The child version APP binds to a family member and enters the family home page (input b3)"}, {"Upon receiving an alarm SMS notification of a fall event in the family home page (input c3)"}, {"After receiving the alarm SMS, the child version APP views the alert message and self-manages emergency situations (input d3)"}, {"In the family home page, the child version APP receives image information of a fall event and enters the fall information page to view images of the elderly person's fall (input e3)"}, {"While viewing the fall situation in the APP, the child self-manages emergency situations (input f3)"}, {"The child version APP supports unbinding and can rebind family members (input g3)"}.

The offspring-focused APP has a finite state machine with an initial state q_0 = "Start APP". The termination states are defined as F = {"Self-Management"}, which represents the service closed or exit the program. The transition function δ_3: $Q_3 \times \Sigma_3 \rightarrow Q_3$ for the children's version APP is mapped as follows and is depicted in Figure 3:

δ_3("Start APP", a3) = "Bind Family Member";

δ_3("Bind Family Member", b3) = "Family Home Page";

δ_3("Family Home Page", c3) = "Alarm SMS";

δ_3("Alarm SMS", d3) = "Self-Management";
δ_3("Family Home Page", e3) = "Fall Image";
δ_3("Fall Image", f3) = "Self-Management";
δ_3("Family Home Page", g3) = "Bind Family Member".

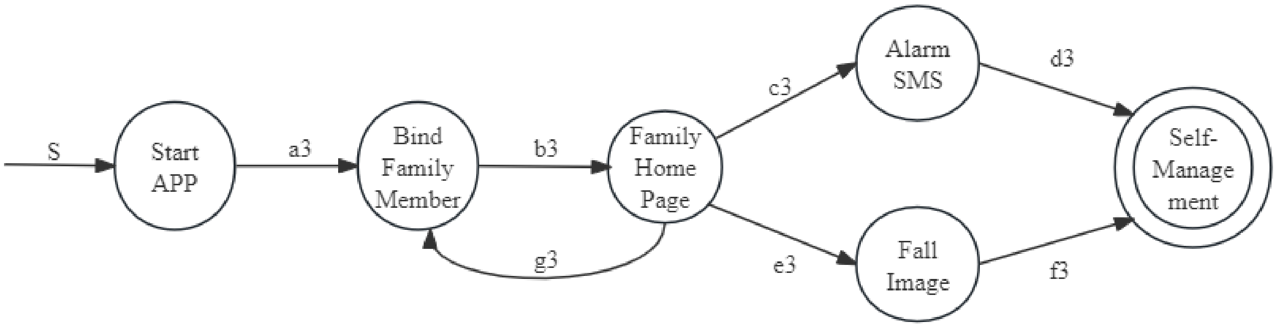

Figure 3. Finite state machine (FSM) diagram of offspring-focused M_3's state transition.

3. Conclusion

Based on the abstract and formalized characteristics of formative language theory courses, this paper employs a case-based teaching method to facilitate students' understanding and mastery of course content. By linking theoretical concepts with practical applications through concrete examples, students are able to see how theories can be applied in real-world practice. Better teaching results can be ultimately achieved in this method by sparking students' enthusiasm for learning and encouraging their more effort into studying.

Acknowledgments

This work was funded by Harbin Institute of Technology (number IDEA10002233).

Conflicts of interests

The authors declare no conflict of interest.

Authors' contribution

Conceptualization, Wei Wang and Yangxuan Cheng; methodology, Wei Wang; software, Xin Sun; validation, Wei Wang, Yangxuan Cheng and Guanglu Zhou; formal analysis, Wei Wang; investigation, Wei Wang; resources, Wei Wang; data curation, Xin Sun; writing—original draft preparation, Wei Wang; writing—review and editing, Yangxuan Cheng and Guanglu Zhou; visualization, Wei Wang; supervision, Ping Yu; project administration, Wei Wang; funding acquisition, Wei Wang. All authors have read and agreed to the published version of the manuscript.

References

[1] Rodger SH, Lim J, Reading S. Increasing interaction and support in the formal languages and

automata theory course. *ACM SIGCSE Bull.* 2007, 39(3):58–62.
[2] Zhao L, Wang X, Qian J. Improving lectures in the course on formal languages and automata theory (In Chinese). *High. Educ. Forum* 2008(3):113–115.
[3] Huai L, Cui R, Yi Z. Exploration and practice of teaching methods for formal language and automata theory course (In Chinese). *Comput. Educ*. 2019(06):106–108.
[4] Pei Z, Tian H, Shu J, Zhou C, Li L, *et al*. Heuristic teaching of the course "formal language and automata (Theory)" based on the improvement of students' autonomous learning ability study reform (In Chinese). *Comput. Knowl. Technol*. 2020, 16(19):106–110.
[5] Zhai Z, Chen J, Li L. Aging in China: General Trends, New characteristies and corresponding policies (In Chinese). *J. Shandong Univ. (Philos. Soc. Sci.)* 2016(3):27–35.
[6] Nie Y. Research on community home-based care for the aged in the new era in China (In Chinese). Jilin University. 2023.
[7] Zhou K. Research on fall detection system for solitary elderly based on wearable equipment (In Chinese). Lanzhou Jiaotong University. 2019.
[8] Wang W, Huang J, Xu Y. Application of finite automata in vehicle management system development (In Chinese). *J. Harbin Univ. Commer. (Nat. Sci. Ed.)* 2012, 28(4):444–446.
[9] Chen F, Li W. Finite automata models of software safety analysis (In Chinese). *J. Northwest Univ.* 2011, 41(1):22–26.
[10] Zhang J. Application of nondeterministic finite automata in braille transcoding (In Chinese). *Comput. Sci.* 2017, 44(1):271–276.